

ROADRACE

By Ian Thompson

Imagine yourself at the wheel of a high speed racing car winding along a treacherous course. To stay on course, you must steer accurately or risk a collision with the side fences. By adjusting the road width and visibility conditions, *Roadrace* can be made as easy or as challenging as you wish.

The road width can be set between 4 and 15 characters, the degree of difficulty changing with different

widths. Visibility can be set to any of four settings. When visibility is good, the car appears high on the screen. This allows a good view of the twisting road ahead. When visibility is poor, the car appears low on the screen allowing only a brief look at the coming road.

After a five step starting light count down the race begins, the twisting and turning road moving continuously on the screen.

The car is steered by the use of the left and right cursor control keys.

The race proceeds until the car crashes off the road. Each collision is considered to terminate one day of the race. After each day, you are shown the distance achieved that day along with the cumulative distance achieved for consecutive days of the race.

Main routines

140- 250 Variable initialisation and graphics display.
300- 420 Accepts road conditions from user.
500- 540 Initialises the road.
600- 650 Determines the next road condition.
700- 750 Updates the car position, determines if crash has occurred.
800-1050 Processes end of race.
1400-1600 Draws next road segment.
2000-2200 Initialises string variables.
3000-3640 Initial graphics display.
4000-4090 Graphics to start race.

Main variables

W Road width.
V Visibility.
M Distance driven on current day.
N Number of days of the race.
T Total distance driven for whole race.
H Elapsed time during race.
L\$,R\$ String characters to move car left, right.
L Position of left side of road.
LC,RC Random value to move road left, right.
EL,ER Leftmost, rightmost allowable road position.
Q\$ User replies.
Z Screen location of car.
RSS,RL\$ Strings to display road segments.
G First address of screen memory.
C\$ Character string for car.

The program occupies 2.8k of memory.

Modifications for TRS-80

The following line modifications will allow the program to run on the TRS-80 Color Computer.

```
160 CR=3:CC=3
210 G=1024
730 IF PEEK(Z+G) < > 144 THEN 800
740 IF PEEK(Z+G+1) < > 144 THEN 800
910 PRINT@480,CHR$(143)
4000 Q=175:K=179
```

The SOUND and COLOR statements must also be changed as appropriate for the TRS-80.

```
0 *****
1 *   R O A D R A C E   *
2 *****
3 *   V Z - 2 0 0 (8K)   *
4 * IAN THOMPSON -COLLARDY *
5 *****
100 SOUND 28,6
140 CLEAR 200
150 LC=0.45
160 CR=3:CC=3
170 L$=CHR$(77):R$=CHR$(44)
200 RC=1-LC
210 G=28672
250 GOSUB 3000
300 GOSUB 3600
310 T=0:N=0
315 CLS:PRINT
320 INPUT"ENTER ROAD WIDTH (4-15)";W
330 W=INT(W):PRINT
340 IF W<4 OR W>15 THEN 310
350 PRINT"VISIBILITY CONDITIONS"
360 PRINT"  1 - TERRIBLE"
370 PRINT"  2 - BAD"
380 PRINT"  3 - FAIR"
390 PRINT"  4 - GOOD"
395 PRINT@280,""
400 INPUT"ENTER VISIBILITY (1-4)";V
410 V=INT(V):GOSUB 2000
420 IF V<1 OR V>4 THEN 395
500 N=N+1:EL=449:ER=478-W:H=0
510 Z=527-64*V:L=463-INT(W/2)
520 FOR J=1 TO 16:PRINT@480,B$;
530 GOSUB 1400:Q$=INKEY$:NEXT
540 PRINT@Z,C$;:GOSUB 4000
600 H=H+1:Q=RND(0):PRINT@480,B$;
610 IF Q>RC AND L<ER THEN 640
620 IF Q<LC AND L>EL THEN 650
630 GOSUB 1400:GOTO 700
640 GOSUB 1600:GOTO 700
650 GOSUB 1500
700 Q$=INKEY$
710 IF Q$=L$ THEN Z=Z-1
720 IF Q$=R$ THEN Z=Z+1
730 IF PEEK(Z+G)<>144 THEN 800
```



```

740 IF PEEK(Z+G+1)<>144 THEN 800
750 PRINT@Z,C$;:GOTO 600
800 FOR J=1 TO 6:Q$=INKEY$
810 PRINT@Z,D$;:SOUND 31,2
820 FOR K=1 TO 10:NEXT
830 PRINT@Z,C$;
840 FOR K=1 TO 10:NEXT:NEXT
900 M=H/50:T=T+M
910 PRINT@480,CHR$(127)
920 PRINT"YOU WENT";M;"KILOMETERS"
925 PRINT"FOR A TOTAL OF";T;"KILOMETERS"
930 PRINT"IN";N;"DAY(S)":PRINT
940 PRINT"HIT <C> - CONTINUE RACE"
950 PRINT"      <R> - RESTART RACE"
960 PRINT"      <Q> - QUIT"
970 Q$=INKEY$
980 IF Q$="C" THEN 500
990 IF Q$="R" THEN 1010
1000 IF Q$<>"Q" THEN 970
1010 PRINT
1020 PRINT"AVERAGE KILOMETERS PER DAY "
1030 PRINT"WAS";T/N;"KM."
1040 IF Q$="R" THEN 310
1050 END

1400 COLOR2:PRINT@L,RS$;:COLOR3:RETURN
1500 COLOR2:L=L-1:PRINT@L,RL$;:COLOR3:RETURN
1600 COLOR2:PRINT@L,RR$;:L=L+1:COLOR3:RETURN
2000 Q=121+CC*16:K=118+CC*16
2010 C$=CHR$(Q)+CHR$(K)
2020 Q=127+CR*16:RS$=CHR$(Q)
2030 FOR J=1 TO W
2040 RS$=RS$+CHR$(128):NEXT
2050 RS$=RS$+CHR$(Q)
2060 Q=119+CR*16:K=120+CR*16
2070 RL$=CHR$(Q)+CHR$(K)
2080 FOR J=1 TO (W-1)
2090 RL$=RL$+CHR$(128):NEXT
2100 RL$=RL$+CHR$(Q)+CHR$(K)
2110 Q=116+CR*16:K=123+CR*16
2120 RR$=CHR$(Q)+CHR$(K)
2130 FOR J=1 TO (W-1)
2140 RR$=RR$+CHR$(128):NEXT
2150 RR$=RR$+CHR$(Q)+CHR$(K)
2160 B$="":FOR J=1 TO 32
2170 B$=B$+CHR$(128):NEXT
2180 D$=CHR$(128)+CHR$(128)
2200 RETURN
3000 W=7:GOSUB 2000:CLS
3010 FOR J=1 TO 15:READ Q
3015 COLOR3
3020 PRINT@Q,RS$;:NEXT
3030 FOR J=1 TO 600:NEXT
3035 COLOR2
3040 RESTORE:FOR J=1 TO 6
3050 READ Q:PRINT@Q+3,C$;
3060 FOR K=1 TO 100:NEXT:NEXT
3070 T$="ROADRACE":FOR J=1 TO 8
3080 READ Q:PRINT@Q+3,C$;
3090 Q$=CHR$(128)+MID$(T$,J,1)
3100 FOR K=1 TO 100:NEXT
3110 PRINT@Q+3,Q$;:NEXT
3120 READ Q:PRINT@Q+3,C$;
3130 SOUND 26,4
3140 FOR J=1 TO 500:NEXT
3160 RETURN
3200 DATA 12,44,77,110,141,172
3210 DATA 205,238,271,304,337
3220 DATA 370,403,436,469
3600 A$=INKEY$:IF INKEY$<>" " THEN 3600
3610 PRINT@448,"HIT ANY KEY TO BEGIN"
3620 Q=RND(0):Q$=INKEY$
3630 IF Q$="" THEN 3620
3640 RETURN
4000 Q=175:K=179
4010 N$=CHR$(Q)+CHR$(Q)+CHR$(Q)
4020 M$=CHR$(Q)+CHR$(K)+CHR$(Q)
4030 Q=7-INT(W/2)-5:K=Q-128
4040 FOR J=K TO Q STEP 32
4045 COLOR4
4050 PRINT@J,N$;:NEXT
4060 FOR J=K TO Q STEP 32
4070 FOR R=1 TO 300:NEXT
4080 PRINT@J,M$;:SOUND 28,4
4090 NEXT:COLOR2
4100 RETURN

```

FOR J=1 TO 500: NEXT